**ZS DATA SCIENCE CHALLENGE - 2019**

Q1. Christiano Ronaldo

Language of Choice = Python3

1. DATA PREP

Isolated test and train set. Almost all data is unclean and needs formatting. Some categorical variables present and some continuous.

Logically decided that columns like **match\_id, team\_id** should not affect the probability of hitting a goal, so such columns were removed.

The columns **lat/lng** was not useful in itself so was split into two separate columns for training.

Duplicate columns like **remainin\_min.1, distance\_of\_shot.1, etc.**  were dropped due to duplicity.

For columns **type\_of\_shot, type\_of\_combined\_shot,** it was realised that they complement each other i.e.if one is null then the other is not null. So the two columns were manipulated as boolean types based on nullness and another column was created to indicate type of shot.

2. EDA

A heatmap was created to find out the correlation between the predictors and the target. It was found out that they aren’t linearly correlated and some form of polynomial or non-linear relationship was being exhibited. It would be an educated assumption that logistic regression wouldn’t be that helpful.

Columns like **area\_of\_shot, shot\_basics** are categorical and logically should impact the target.

Imputers were created to impute almost all columns based on mean, median and frequency.

3. MODELLING

Several imports were made for several models, namely **XGBoost, DecsionTreeClassifier, SVC** and **RandomForestClassifier**.

A pipeline was created all the above model part of it. Data was passed to each model after scaling using *StandardScaler()*.

Using 5 fold cross-validation on the training set led me to see that XGBClassifier performed the best with MAE as the metric for judging performance.

The next step was to hyperparameter tuning for XGBClassifier. Using GridSearchCV, the depth and n\_estimators were determined for the best possible score. It was found that depth of 5 and 50 estimators work best.

Then the full training set was taken for final building and making predictions on the test set.

4. CONCLUSION

The result is poor. But it was found that scaling does improve by slight margin. One-hot encoding increases the size of matrix too much and is computationally expensive. Hence LabelEncoding was done which might misinform the model. So pros and cons. Some columns were imputed with means and some with medians based on EDA.